Möbius Voting For Surface Correspondence

Yaron Lipman   Thomas Funkhouser

Princeton University

Abstract

The goal of our work is to develop an efficient, automatic algorithm for discovering point correspondences between surfaces that are approximately and/or partially isometric.

Our approach is based on three observations. First, isometries are a subset of the Möbius group, which has low-dimensionality – six degrees of freedom for topological spheres, and three for topological discs. Second, computing the Möbius transformation that interpolates any three points can be computed in closed-form after a mid-edge flattening to the complex plane. Third, deviations from isometry can be modeled by a transportation-type distance between corresponding points in that plane.

Motivated by these observations, we have developed a Möbius Voting algorithm that iteratively: 1) samples a triplet of three random points from each of two point sets, 2) uses the Möbius transformations defined by those triplets to map both point sets into a canonical coordinate frame on the complex plane, and 3) produces "votes" for predicted correspondences between the mutually closest points with magnitude representing their estimated deviation from isometry. The result of this process is a fuzzy correspondence matrix, which is converted to a permutation matrix with simple matrix operations and output as a discrete set of point correspondences with confidence values.

The main advantage of this algorithm is that it can find intrinsic point correspondences in cases of extreme deformation. During experiments with a variety of data sets, we find that it is able to find dozens of point correspondences between different object types in different poses fully automatically.

1 Introduction

Finding correspondences between a discrete set of points on two different surface meshes is a fundamental problem in computer graphics, geometric processing, and medical imaging. Among its many applications are shape interpolation, attribute transfer, surface completion, statistical shape modeling, symmetry analysis, shape matching, and deformable surface tracking.

For many of these applications, the input meshes represent different objects in different poses, and thus alignment of extrinsic shapes is insufficient. However, the intrinsic shapes for objects of the same class are often approximately isometric, and sometimes they are composed of large parts that are nearly isometric. For example, this is the case for the dog and the wolf shown in Figure 1, as well as for cloth deformations (perfect isometries), faces, and surfaces
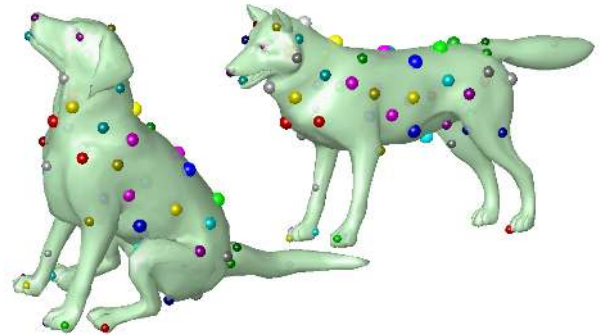

Figure 1: Correspondences found automatically between a dog and a wolf. Corresponding points are shown as spheres of the same color with a radius proportional to the confidence of the correspondence.

of brains and other anatomical organs. The goal of our work is to develop efficient algorithms for discovering dense sets of point correspondences for these shapes automatically.

This problem is challenging for two reasons. First, the space of possible correspondences is very large ($O(N!)$ for N input points), and thus strategies for searching this space with the hope of finding the correspondence with minimal deformation error are either very slow or resort to heuristics. Second, even if it were possible to consider all potential correspondence sets, it would be difficult to compute an appropriate deformation error (deviation from isometry) for each one. Therefore, previous methods have relied upon humans to provide initial landmark correspondences, work only on examples with similar local shape features, compute approximate deformation errors, and/or take long computation times.

In this paper, we propose an algorithm that we call Möbius Voting. The key observation is that the space of isometries between simply-connected surfaces is contained within the Möbius group, which has low dimensionality. For genus zero surfaces (sphere topology), the Möbius group has six degrees of freedom; and for patches with boundaries (disc topology), it has only three. Thus, defining the entire mapping between two isometric surfaces is possible with just three point correspondences. Moreover, there is a Conformal Flattening (Uniformization) operator (based on Pinkall and Polthier's non-conforming conjugate harmonic maps [1993]) that takes a mid-edge representation of the mesh onto a 2D canonical domain defined by three points, where: 1) the discrete conformal error of the mapping is zero, 2) the Möbius transformation defined by the three point correspondences can be computed in a closed form, as a rational linear function in the complex plane, 3) the Möbius transformation can be applied to any point in the complex plane by computing a simple rational function that is fast to compute, and 4) deviations from an isometric mapping can be modeled with a simple function based on transportation-type distances.

The main implication of these observations is that it is possible to design an algorithm that finds a dense set of correspondences between nearly isometric meshes in polynomial-time. At the core of our algorithm is a Hough-style voting scheme where three random points are repeatedly sampled from each of two meshes and used to define Möbius transformations that map them to a shared canonical

Möbius-invariant representation. Intuitively, if these transformations align many points in the two sets (Figure 3), then the three sampled point pairs are probably true correspondences, as are all the ones that are aligned (are mutually closest). So, we log votes for all these potential point correspondences with magnitude inversely related to how well they conform to an isometric mapping. The result after $O(N^3)$ votes is a "fuzzy correspondence matrix," where each element $C(i,j)$ provides a score for how likely the i-th point of one set is in correspondences with the j-th point of the other. We transform this fuzzy matrix into a discrete permutation matrix by a greedy algorithm, which provides a discrete set of point correspondences and a confidence score for each one as output.

The main advantage of this algorithm is that it finds partial and approximate point correspondences automatically for a wide class of deformations. Rather than relying upon local shape descriptors, thresholds, and other heuristics to prune bad correspondences, it leverages invariants of the global conformal structures of the meshes to find the set of correspondences that is most consistent with isometric mappings between large regions of the surfaces. During experiments, we find that this strategy is able to find dozens of point correspondences between meshes of several types fully automatically (e.g., Figure 1).

The main contribution of this paper is the idea of leveraging the low-dimensionality of the Möbius group in an automatic correspondence-finding algorithm. In order to capitalize on this idea, we had to make several smaller contributions. First, we employ a mid-edge flattening scheme and prove that it is robust to holes and boundaries (Section 5). Second, we suggest a variant of the Generalized Hough Transform that finds partial sets of correspondences by voting directly into a fuzzy *correspondence matrix* – e.g., rather than voting for transformations and then clustering (Section 6). Third, we define a transportation distance measure of how far the mapping defined by a set of point correspondences is from a perfect isometry (Section 7). Finally, we provide results of experiments on several databases and qualitative results.

2 Previous Work

Finding correspondences between two sets of discrete of points sampled from a surface is a classical problem, and thus there is a large literature on the subject. Previous approaches can be characterized by: 1) the method used to measure deformation error, and 2) the strategy used to search for correspondences. In all cases, the goal is to find the (possibly partial) set of correspondences that minimize the deformation error.

Measuring Deformation Error: Most work on non-rigid alignment of meshes has focused on cases where small deformations are expected – e.g., for registration of poorly calibrated range scans [Brown and Rusinkiewicz 2007; Li et al. 2008], for statistical analysis of similar objects in the same pose [Allen et al. 2003], or for medical imaging [Audette et al. 2000]. For these problems, low-frequency deformation models are sufficient (e.g., thin-plate splines [Brown and Rusinkiewicz 2007]). For larger scale deformations, such as those found in template fitting [Pauly et al. 2005], deformation transfer [Sumner and Popovic 2004], intrinsic symmetry analysis [Ovsjanikov et al. 2008], and other applications where objects have different poses and/or different local shapes, more flexible deformation models are required. Some previous work has relied upon iterative optimization of nearly rigid deformations [Allen et al. 2003; Pauly et al. 2005; Mitra et al. 2007; Ovsjanikov et al. 2008], while others have explicitly defined error functions that measure global extrinsic deformations [Botsch and Sorkine 2008]. For example, [Eckstein et al. 2007] use a pseudo-Hausdorff distance to define matching between surfaces, where their optimization process is defined via generalized gradient flow.
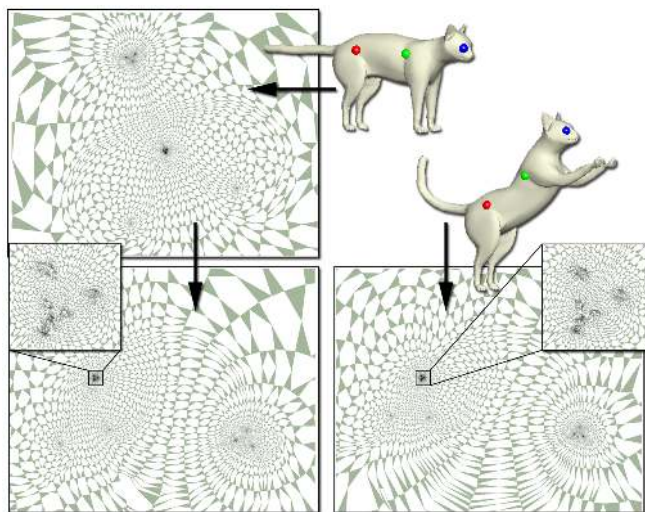

Figure 3: Fixing three corresponding pairs of points (red, green, and blue) on two different poses of the cat defines a Möbius transformation which applied to the flattening of the first pose (top-left) and yields the flattening at the bottom-left. Note that after applying this transformation the two flattenings (bottom row) look similar.

Several methods aim to define a deformation error that models deviations from isometry. For example, [Bronstein et al. 2006] use a variant of the Gromov-Hausdorff distance. This is a nice mathematical formulation, but relies upon repeatedly computing geodesic distances between arbitrary pairs of points on a mesh, and so it is expensive to compute. A less expensive approach is to embed the mesh in a (possibly higher dimension) space such that the Euclidean distance between two embedded points is an approximation of the geodesic distance on the original mesh [Elad and Kimmel 2003; Jain et al. 2007; Ovsjanikov et al. 2008]. Then deformation error can be measured with a rigid model in the embedding space. The main difficulty in this approach is achieving a good isometry-preserving embedding. Indeed, it is well-known that, in general, a (non-flat) surface cannot be mapped without error to the Euclidean plane. So, even in the perfect case, where the two input surfaces are perfect isometries, their embedding is not congruent; and, hence rigid alignment in the embedding space is not the same as isometry on the original mesh.

Other methods have used human-provided landmark points and/or correspondences found with local shape descriptors to define a mapping between two surfaces, and then measure deviations of surface properties (e.g., curvature) at points aligned in the map. For example, [Wang et al. 2007] use point constraints predicted by spin images [Johnson and Hebert 1999] to establish a sparse set of correspondences and then use least-squares conformal mapping to create a cross-parameterization. In later work, [Zeng et al. 2008] further elaborate this direction by cutting the surfaces into patches, given user-defined boundary correspondence, and combine several conformal mappings. Furthermore, they measure deformation error by an integral of the differences between conformal factors and curvatures over the domain of the map. Here again, the feature point constraints are either user-defined or produced by local shape descriptors, such as SIFT [Lowe 2004]. The drawback of these approaches is that they rely upon a sparse set of feature correspondences derived only from local shape properties, and they do not take into consideration *global structure* during this process. Thus, they would not work well when the local shapes of two surfaces differ greatly.
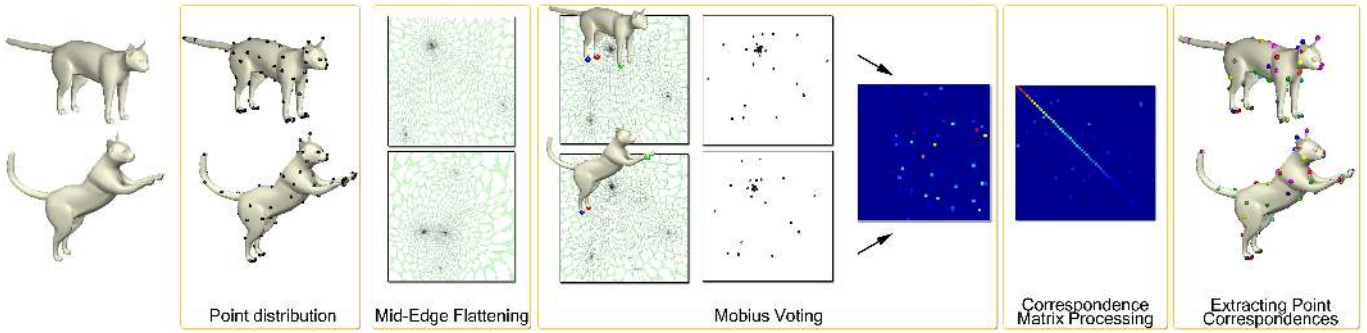
Figure 2: The outline of our algorithm as described in the Overview Section 3. Note that the Mid-edge flattenings displayed in the voting stage are not used in the algorithm and put here for reference.

Searching for Correspondences: There are several ways to find the set of correspondences that minimizes a deformation error. The most general method is to perform a combinatorial search of the correspondence space. Since $O(N!)$ different correspondence permutations are possible amongst two point sets of size N (without even allowing outliers), brute force search is intractible. Thus, several researchers have investigated search strategies based on branch-and-bound [Gelfand et al. 2005], priority-driven search [Funkhouser and Shilane 2006; Zhang et al. 2008], belief propagation [Anguelov et al. 2004], and integer quadratic programming [Berg et al. 2005]. However, still, they usually require thresholds to prune poor correspondences, and are slow.

A faster approach is based on Iterative Closest Points (ICP) [Besl and McKay 1992]. This greedy method finds an initial set of correspondences based on closest points (possibly in an embedding space), deforms the surface according to those correspondences, and repeats until convergence. Although it has more commonly been used for registration of rigid surfaces [Rusinkiewicz and Levoy 2001], ICP has also been applied for large number of non-rigid deformation problems [Allen et al. 2003; Bronstein et al. 2006; Brown and Rusinkiewicz 2007; Huang et al. 2008; Pauly et al. 2005]. In particular, [Mitra et al. 2007] and [Huang et al. 2008] use a variant for finding correspondences across extreme deformations, and [Bronstein et al. 2006] use it for finding correspondences between surfaces that differ by an isometry. The problems with this approach are that it relies upon an initial guess for either the correspondences or the deformation, and it finds a local minimum. Thus, it is not fully automatic and can produce suboptimal solutions.
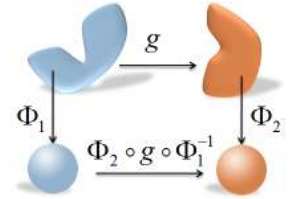
Several authors cast the (rigid and non-rigid) correspondence problem into a fuzzy linear assignment problem [Gold et al. 1998; Chui and Rangarajan 2003]. In this approach, the problem is written in a form of a functional which contains both the sought for correspondences (which can contain continuous values $[0-1]$) and the deformation error function. Then, this functional is minimized by interleaving between optimizing for correspondences and optimizing for deformations. In our work, we will also be using correspondences matrices with real coefficients, but these matrices will be constructed by a randomized voting algorithm.

Perhaps the state-of-the-art algorithm for automatic correspondence finding for extreme deformations is the recent paper by [Zhang et al. 2008]. They use a priority-driven strategy to search the combinatorial space of correspondences. For each proposed correspondence set, they deform the mesh with the method in [Lipman et al. 2005] and measure distortion with an error functional defined in [Kraevoy and Sheffer 2006]. They are able to find 5-10 correspondences between meshes with significant deformations.

3 Overview

Given two meshes, $\mathcal{M}_1$ and $\mathcal{M}_2$, as input, our goal is to establish a set of correspondences between points on those surfaces such that the mapping between them is close to an isometry.

The key observation which makes this problem tractable is best understood in the smooth case: isometries are a subset of a larger group of conformal mappings. The Uniformization Theorem states that any genus zero surface (sphere topology) can be mapped conformally (with angles preserved) to the unit sphere. Therefore, any isometry between genus zero surfaces gives rise to a one-to-one and onto conformal map from the unit sphere to itself (depicted in the figure on the right where g denotes the isometry and Φ_1, Φ_2 are the uniformization maps). The group of all one-to-one and onto mappings of a sphere to itself is well understood and is called the Möbius group. This group has six degrees of freedom, and so fixing three distinct points on each sphere defines a Möbius map uniquely.[1]

This observation allows us to develop a polynomial-time algorithm for finding point correspondence between nearly-isometric surfaces as shown in Figure 2. The algorithm iteratively: 1) samples three points from each mesh, 2) computes the Möbius transformations that align those three point pairs in a canonical domain, 3) transforms all (sampled) points from both meshes by those Möbius transformations, and 4) measures deformation error between mapped points (deviation from isometry). The computed deformation errors are accumulated in a fuzzy correspondence matrix, which can be analyzed to determine a consistent set of discrete correspondences. This polynomial-time algorithm is (theoretically, in the smooth case) guaranteed to find the optimal set of correspondences for perfect isometries and extends well to near isometries.

Implementing this algorithm requires addressing three non-trivial issues. The first is computing Möbius transformations. Given three points directly on two meshes, it is not clear how to calculate the Möbius transformation defined by them. So, we need to map the meshes to a canonical domain where the Möbius transformation have a simple closed-form formula. For this step, we employ a mid-edge Conformal Flattening (Uniformization) based on Pinkall

---

[1]For two surfaces with disc topology, the Möbius group has three degrees of freedom, and so only one corresponding point and a direction are required.

and Polthier's non-conforming conjugate harmonic maps [Pinkall and Polthier 1993; Polthier 2005]. This mid-edge mapping scheme has a zero discrete-conformal error, maps the mesh onto a 2D domain, and provides a way to compute Möbius transformations from three point pairs in a closed form, as rational linear functions in the complex plane (Section 5).

The second issue is how to measure deformation errors between two discrete point sets for a given triplet of correspondences. If indeed the triplet are true correspondences and the point sets are truly isometric, then the Möbius transformation that aligns the three points will also align every other point with its true correspondence in the canonical domain. However, if the points are not true correspondences, or if the surfaces are only approximately or partially isometric, then we need a quick-to-compute measure of the deformation error that grows continuously as the meshes deviate from perfect isometry. For this, we utilize a transportation-type distance based on Euclidean distances between mutually closest point in the canonical domain (Section 7).

The third issue is to devise a search procedure that finds partial and approximate correspondences robustly. Since the Möbius group has low-dimensionality, several common approaches are possible. For this project, we implemented an algorithm based on the Generalized Hough Transform [Ballard 1981]. It samples triplets of point correspondences from two discrete point sets, computes the Möbius transformation that aligns them in the canonical flattened space, computes the deformation error for all other mutually closest points, and accumulates "votes" for correspondences between mutually closest points based on the deformation error. In order to avoid issues of biased sampling and clustering, we employ a variant that votes for correspondences rather than transformations, building a "fuzzy correspondence matrix" that models the probability that two points correspond (Section 6). After many votes are cast, this continuous-valued matrix is converted to a binary permutation matrix using a greedy max-row-column algorithm, which provides the final set of predicted point correspondences with confidence values (Section 8).

Details of the algorithms for each step of the process appear in the following five sections. Section 9 presents results of experiments aimed at testing how well the algorithms perform for a variety of object types, and Section 10 provides discussion of conclusions, limitations, and topics for future work.

4 Point samples

Given two meshes, $\mathcal{M}_1$ and $\mathcal{M}_2$, our first step is to sample a discrete set of N points from each one, $\Sigma_1 \subset \mathcal{M}_1$, $\Sigma_2 \subset \mathcal{M}_2$. Since these points sets provide the candidates for potential correspondences, they should be invariant to isometries and sampled from "consistent" regions of the surface (e.g., tips of noses, toes, etc.). Also, since the points will be used to vote for correspondences, each point should carry the same "importance."

To construct such a point-sample set, we first take all points p_ℓ that are local maxima of Gauss curvature. These points form a discrete set, invariant to isometries, and therefore often have correspondences in other objects of the same type. Then we spread the rest of the N sample points Σ_1 uniformly such that each of these points p_k represents a cell Ω_k, and all the cells have approximately the same area ($area(\Omega_{k_1}) = area(\Omega_{k_2}), k_1 \neq k_2$). For this end we take a spread of points with the Farthest Point Algorithm (FPS) [Eldar et al. 1997]. Starting from the set of Gauss local maxima, we repeatedly add the most distant (geodesically) point from all the points we have collected so far. In case no Gauss maxima is taken this process provides an approximately uniform sampling where each point represents a surface patch of approximately equal size,

and where there are upper and lower bounds on the maximal empty geodesic disc and closest distance between two samples. In our current implementation, we use an approximate algorithm based on Dijkstra's Shortest path algorithm to compute geodesic distances, and a smoothed version of the angle-deficit formula [Desbrun et al. 2002] to approximate Gauss curvature. In all the examples of this work we used around 100 sample points.

5 Mid-edge Uniformization

Our second step is to map each mesh (and its associated point set) into a domain where computing the Möbius transformation defined by three point correspondences is efficient. While in the smooth case the Uniformization (or conformal flattening) of a surface is always possible, this is not the case in the discrete case. That is, it is not generally possible to flatten a mesh such that each triangle undergoes exact similarity. Previous methods that have mapped meshes to spheres [Praun and Hoppe 2003; Sheffer et al. 2004] or planes [Hormann et al. 2007] introduce error in the sense that each triangle is flattened by a near-similarity. Since we are not aiming to solve the parameterization problem and our main goal is getting a representation which preserves the conformal structure of the mesh, we consider the more flexible *mid-edge* structure. That is, we employ a mid-edge Conformal Flattening (Uniformization) based on Pinkall and Polthier's non-conforming conjugate harmonic maps [Pinkall and Polthier 1993]. This mid-edge flattening transforms each triangle (face) by an exact similarity and is only continuous at the mid-edges. Therefore, it can be seen as a flattening of the mid-edge mesh (to be defined shortly) onto a 2D domain. This provides a way to compute Möbius transformations from three point pairs in a closed form. Besides that it has a nice theoretic property regarding boundaries (and holes) of the original mesh.

Our description of this mapping begins with a definition of a mid-edge mesh. Denote a triangular mesh by $\mathcal{M} = (V, E, F)$, where $V = \{v_i\}$, $E = \{e_{i,j}\}$, $F = \{f_{i,j,k}\}$ denote the vertices, edges and faces respectively. Then the mid-edge mesh $\mathbf{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$ is defined as follows: The vertices of $\mathbf{M}$, that is $\mathbf{V} = \{\mathbf{v}_r\}$, are defined as the mid-edge points of the mesh $\mathcal{M}$. So, for each edge $e_{i,j} \in E$ corresponds a vertex $\mathbf{v}_r \in \mathbf{V}$. Next, for each face $f_{i,j,k} \in F$ corresponds a unique face $\mathbf{f}_{r,s,t}$ which consists of mid-edge points of $f_{i,j,k}$ with the same orientation. We further denote by $\mathbf{e}_{sr}$ the edge in the mid-edge mesh between $\mathbf{v}_s$ and $\mathbf{v}_r$. It should be noted that the mid-edge mesh is not a manifold mesh in the sense that each edge has only one face touching it. See the image on the right for a mesh patch (black vertices and edges) and its corresponding mid-edge mesh (mid-edge vertices in red, and faces in purple).
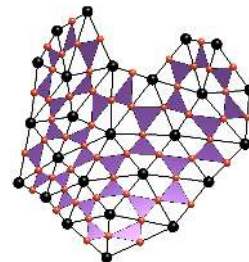
The most important property of the mid-edge mesh in our context is that it has more flexibility than the original mesh and can be flattened easily onto the plane with zero discrete conformal error (Algorithm 1). That is, each of its faces undergos a similarity transformation. This is done by defining two piecewise linear discrete conjugate harmonic functions on the mesh: $u(\cdot)$ which is continuous along the edges of the mesh and $u^*(\cdot)$ which is only continuous through the mid-edges of the mesh. Then, the mapping of each mid-edge vertex $\mathbf{v}_r$ into the complex plane is defined by $\Phi(\mathbf{v}_r) = u(\mathbf{v}_r) + \mathbf{i}u^*(\mathbf{v}_r) \in \mathbb{C}$, See Figure 4. This construction can be understood from the work of Pinkall and Polthier [1993], and later work by Polthier [2005] as follows. Let us define discrete har-
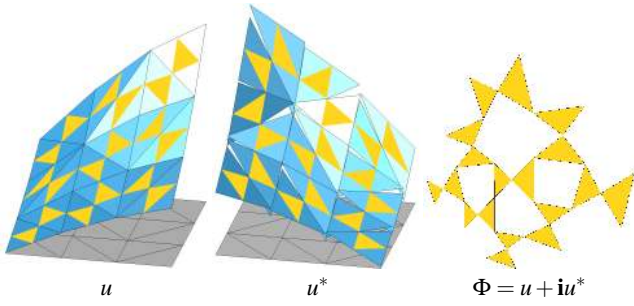
u u^* $\Phi = u + \mathbf{i}u^*$

Figure 4: The mid-edge flattening based on Pinkall and Polthier's discrete harmonic function and its conjugate. The mid-edge flattening is done by defining two scalar functions over the mesh, drawn here as a 2D wireframe: u which is a discrete harmonic piecewise-linear function and its conjugate u^*. Each mid-edge vertex $\mathbf{v}_r$ is mapped to the plane via $\Phi(\mathbf{v}_r) = u(\mathbf{v}_r) + \mathbf{i}u^*(\mathbf{v}_r)$. The mid-edge mesh is shown in yellow.

monic map on the mesh $\mathcal{M}$ using the cotangent-formula. That is, consider a continuous piecewise linear function $u(\cdot) = \sum_i u_i \phi_i(\cdot)$, where $\phi_i(\cdot)$ is the linear hat function on the mesh which is one at v_i and zero on v_j, $j \neq i$, and u_i are the function values at the vertices v_i. Then $u(\cdot)$ is defined as discrete harmonic if it satisfies the following set of linear equations:

$$\sum_{j \in \mathcal{N}(i)} (u_i - u_j)(\cot \alpha_{ij} + \cot \beta_{ij}) = 0 , \text{ for all } v_i \in V \quad (1)$$

where $\mathcal{N}(i)$ denotes the indices of the neighboring vertices of i and the angles α_{ij}, β_{ij} are the angles supporting the edge $e_{i,j}$ in the two faces meeting at this edge.

Pinkall and Polthier define the conjugate discrete harmonic function u^*, by observing that there exists a function that is linear on each face, continuous through the mid-edge points and in each face it is the conjugation (in the complex-analytic sense) of u. Furthermore, it is discrete harmonic in the space of non-conforming piecewise linear finite-element, and it is fixed up to a constant factor. Given the above observations, it is rather simple to calculate the conjugate function u^* by integrating its gradient, defined by (positive) rotation of $\pi/2$ of the gradient of u on each triangle [Polthier 2005]:

$$u_r^* - u_s^* = \frac{1}{2}\left((u_i - u_j)\cot\theta_k + (u_k - u_j)\cot\theta_i\right), \quad (2)$$

where u_r^*, u_s^* are the values of u^* on the mid-edge vertices $\mathbf{v}_r, \mathbf{v}_s$ located at the edges $e_{i,j}, e_{j,k}$ (respectively), and θ_k is the angle at vertex v_k, and similarly θ_i, as depicted in the image to the right.
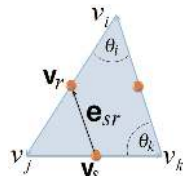
So, setting one value u_r^* arbitrarily and traversing the mesh using the above formula results in the conjugate discrete harmonic function u^* on the mid-edges, then defined linearly in each face of the mesh $\mathcal{M}$.

Since our mesh has sphere topology, when flattened to the complex plane one point should go to infinity. We denote the complex plane with infinity added to it as a point to be the *extended complex plane* and mark it by $\hat{\mathbb{C}}$. So, we need to pick a point which goes to infinity. We will take that point to be inside a face which we will cut off from the mesh. This face is defined by having a vertex which has minimal average geodesic distance measured to all the other vertices. Choosing different cut-face would lead to flattening which
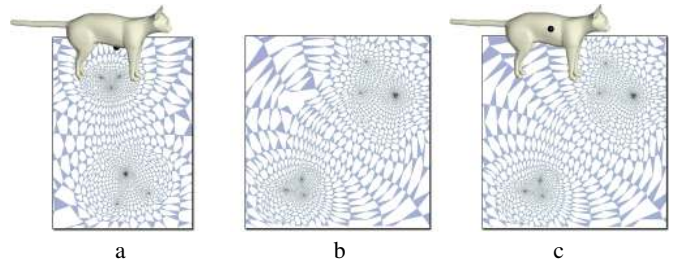

Figure 5: Invariance of the flattening to the choice of cut-face: (a) and (c) are two different mid-edge flattening of the same mesh based on different choices of poles/cut-faces (black points). (a) and (c) are connected by a Möbius transformation: (b) depicts (a) after appropriate Möbius transformation applied to it. Note the resemblance between (b) and (c).

is approximately a Möbius transformation of the former one, see Figure 5. The linear system (1) which defines u, has only the constant solution in its kernel, so to produce a non-constant u we set its value to zero and one at two of the vertices of the cut-face, and delete their corresponding equations from that system (Algorithm 1).

Meshes with holes/boundaries. The mid-edge flattening has an interesting property (proved in the appendix) which allows flattening of meshes with holes and/or boundaries (see Figure 6):

Theorem 5.1 *When mapping the mid-edge mesh* $\mathbf{M}$ *of a mesh* $\mathcal{M}$ *with boundaries using a discrete harmonic and conjugate harmonic pair, the mid-edge vertices of each connected boundary part are mapped exactly onto a line segment parallel to the real axis.*

This result, which is proved in the Appendix, is unique to the mid-edge construction. Furthermore, this result is an exact analog of the situation in the smooth case, see [Springer 1981]. Later on, we will employ one-to-one and onto mappings of the extended complex plane (flattened space) into itself. This property ensures that holes in the meshes are closed in the flattened space so the corresponding mappings between the mid-edge meshes can be thought of as one-to-one and onto.

Figure 6: Mid-edge uniformization of a disc type hand model. Note the boundary mid-edge vertices lies exactly on a horizontal segment, and the three unconnected mid-edge vertices - these are connecting the missing mid-edge face which has infinity in its interior.

6 Möbius Voting

Our third step takes as input two point sets, Σ_1 and Σ_2, and a flattening map for each into the complex plane, $\Phi_1 : \mathbf{M}_1 \to \hat{\mathbb{C}}$ and $\Phi_2 : \mathbf{M}_2 \to \hat{\mathbb{C}}$, and it outputs a fuzzy correspondence matrix, $C = (C_{k,\ell})$, where the value $C_{k,\ell}$ indicates the confidence of the k-th point sample in Σ_1 being in correspondence with the ℓ-th point sample in Σ_2.

```
Input: mesh (V, E, F)
       Corresponding mid-edge mesh (V, E, F)
       A face to cut f_c = {i_c, j_c, k_c} ∈ F
Output: Planar embedding of mid-edge vertices Φ : V → Ĉ

/* Discrete harmonic function u(·)              */
Solve the sparse linear system: u_{i_c} = -1, u_{j_c} = 1,

    Σ      (u_i - u_j)(cot α_ij + cot β_ij) = 0 ,  v_i ∈ V \ v_{i_c}, v_{j_c}.
  j∈N(i)

/* Conjugate harmonic function u*(·)            */
Set arbitrary value (zero) at one mid-edge vertex: u*_0 = 0
Traverse the mid-edge mesh and for each visited edge e_{sr} in
face f_{i,j,k} set the value u*_r by:

    u*_r - u*_s = 1/2 ((u_i - u_j) cot θ_k + (u_k - u_j) cot θ_i)

/* The planar embedding                         */
For mid-edge vertex v_r = (v_i+v_j)/2 Set Φ(v_r) = (u_i+u_j)/2 + i u*_r.
```

Algorithm 1: Mid-edge flattening based on Pinkall and Polthier's discrete harmonic and conjugate harmonic functions. See Section 5 for notation explanation.

```
Input: points Σ_1 = {z_k} and Σ_2 = {w_ℓ}
       number of iterations I
       minimal subset size K
Output: correspondence matrix C = (C_{k,ℓ}).

/* Möbius voting                                 */
while number of iterations < I do
    Random z_1, z_2, z_3 ∈ Σ_1.
    Random w_1, w_2, w_3 ∈ Σ_2.
    Find the Möbius transformations m_1, m_2 s.t.
        m_1(z_j) = y_j, m_2(w_j) = y_j, j = 1, 2, 3.
    Apply m_1 on Σ_1 to get z̄_k = m_1(z_k).
    Apply m_2 on Σ_2 to get w̄_ℓ = m_2(w_ℓ).
    Find mutually nearest-neighbors (z̄_k, w̄_ℓ) to formulate
    candidate correspondence c.
    if number of mutually closest pairs ≥ K then
        Calculate the deformation energy E(c)
        /* Vote in correspondence matrix
           */
        foreach (z̄_k, w̄_ℓ) mutually nearest-neighbors do
            C_{k,ℓ} ← C_{k,ℓ} + 1/(ε+E(c)/n).
        end
    end
end
```

Algorithm 2: Möbius Voting for point correpondences.

The algorithm is quite simple (pseudo-code is shown in Algorithm 2). It first projects every point of $\Sigma_1 = \{p_k\}$, $\Sigma_2 = \{q_\ell\}$ onto the complex plane (using closest mid-edge vertex) to form the planar point samples $z_k := \Phi_1(p_k)$ and $w_\ell := \Phi_2(q_\ell)$, respectively. We will denote these planar sets also as Σ_1 and Σ_2. Then, it iteratively samples random triplets of points from each of Σ_1 and Σ_2, computes the interpolating Möbius transformations that align those triplets in a common domain, transforms all other points of Σ_1 and Σ_2 into that domain, and then computes a deformation error based on how well mutually closest points align in that domain (Section 7). Each such iteration produces "votes" for correspondences with magnitude based on the inverse of the deformation error, which are accumulated in a fuzzy correspondence matrix, C.

This algorithm is efficient for two reasons. First, there is a closed form solution for the unique Möbius transformation that interpolates two triplets of points in the flattened space. A Möbius transformation is a one-to-one and onto map of the extended complex plane $\hat{\mathbb{C}}$ (or equivalently the sphere) to itself defined as the following linear fractional function in the complex variable $z \in \hat{\mathbb{C}}$:

$$ m(z) = \frac{az+b}{cz+d}, \tag{3} $$

where $a, b, c, d \in \mathbb{C}, ad - bc \neq 0$. It can be shown that Möbius transformations form a *group* under composition and that they consist of *all* one-to-one and onto mapping of $\hat{\mathbb{C}}$ (or the sphere) to itself. A Möbius transformation can be represented by the 2x2 matrix of its coefficients $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with non vanishing determinant.

The coefficients of the composition of two Möbius transformations can be calculated by regular matrix multiplication of the corresponding coefficient matrices.

Another interesting property of the Möbius group is that it is 3-transitive, that is, it can interpolate three distinct points uniquely: Indeed, given two triplets $z_1, z_2, z_3 \in \hat{\mathbb{C}}$ and $y_1, y_2, y_3 \in \hat{\mathbb{C}}$ we look for $a, b, c, d \in \mathbb{C}$ such that $y_j = m(z_j)$, $j = 1, 2, 3$. This can be achieved by taking:

$$ \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} y_2 - y_3 & y_1 y_3 - y_1 y_2 \\ y_2 - y_1 & y_1 y_3 - y_3 y_2 \end{pmatrix}^{-1} \begin{pmatrix} z_2 - z_3 & z_1 z_3 - z_1 z_2 \\ z_2 - z_1 & z_1 z_3 - z_3 z_2 \end{pmatrix}, \tag{4} $$

which is merely composition of two Möbius transformations. Note that the transformation which corresponds to $\begin{pmatrix} z_2 - z_3 & z_1 z_3 - z_1 z_2 \\ z_2 - z_1 & z_1 z_3 - z_3 z_2 \end{pmatrix}$ takes z_1, z_2, z_3 to $0, 1, \infty$ (respectively).

Second, Eq. (3) can be used as a closed-form formula for the Möbius transformation of any other point in the flattened space, $z \in \hat{\mathbb{C}}$, to its image, $y = m(z) \in \hat{\mathbb{C}}$.

With these two equations and a suitable deformation error measure, the basic components of the algorithm are in place. However, there are three interesting aspects of our implementation that are worth noting.

First, rather than map Σ_1 to Σ_2 directly, we map both to a canonical domain. This means that we set three constant points in the extended complex plane which form an equilateral (scale and position do not matter), for example set $y_j = e^{\mathbf{i}\frac{2\pi}{3}j}$, $j = 1, 2, 3$. Then, we transform each randomized triplet $z_1, z_2, z_3 \in \Sigma_1$ and $w_1, w_2, w_3 \in \Sigma_2$ to these three constant points y_1, y_2, y_3 using Eq. (3),(4). The advantage in doing so rather than mapping directly z_1 to w_1, z_2 to w_2 and z_3 to w_3 is that mapping to a fixed canonical domain produce an invariant setting which does not depend on the specific initial flattenings Φ_2, Φ_1 of the surfaces. Let us also note that this choice of constant points y_1, y_2, y_3 is arbitrary and taking other points will also work. Note for example, that choosing the triplet $0, 1, \infty$ is equivalent to taking the *cross-ratio* $(z - y_2)(y_1 - y_3)/(z - y_3)(y_1 - y_2)$.

Second, rather than voting for transformations, we vote for correspondences. That is, we maintain a non-negative real matrix $C = (C_{k,\ell})$ with dimensions $|\Sigma_1| \times |\Sigma_2|$. During each iteration, after two point sets, Σ_1 and Σ_2, are mapped into the canonical do-

main by the Möbius transformation defined by $z_1, z_2, z_3 \in \Sigma_1$ and $w_1, w_2, w_3 \in \Sigma_2$, we find mutually closest point correspondences and add them to a set of candidate correspondences c. Note that the pairs of points used to define the Möbius transformations for this iteration, $(z_1, w_1), (z_2, w_2), (z_3, w_3)$, are always in this set (since we defined the Möbius Transformation to take these points to a fixed set of positions, y_1, y_2, y_3). If this corresponding set (number of pairs found) is larger than a threshold K, that is if $n > K$, where n is the number of correspondences in c, we compute the averaged cost of that correspondence $\mathbf{E}(c)/n$ (Section 7), and add $1/(\varepsilon + \mathbf{E}(c)/n)$, where ε is some small number, to all entries belonging to pairs of corresponding points in c. That is, $C_{k,\ell} = C_{k,\ell} + 1/(\varepsilon + \mathbf{E}(c)/n)$ for all (z_k, w_ℓ) in c.

Third, we accumulate votes only for samples where the size of the corresponding set (number of mutually closest pairs found) is larger than a threshold K. This check assures that we only consider sets of corresponding points that both represent large fractions of the surfaces, and thus guarantees that the candidate correspondences c capture metric features (clusters) well and do not miss any significant parts (e.g., with large area). In all our experiments, we set K to be 40% of the number of points samples.

The advantage of voting for correspondences (in comparison to voting for transformations, which is the usual approach for Generalized Hough Transformations) is that it avoids the problems that arise when uniform sampling of points leads to non-uniform sampling of the transformation space. Also, it discovers partial sets of correspondences automatically – there is no need to devise an algorithm to group correspondences based on clusters of votes in transformation space. Rather, the fuzzy correspondence matrix reveals isometric part structure of the two meshes directly.

7 Measuring Intrinsic Deformation Error

This section defines a deformation error that can be used to provide a magnitude for votes when building the fuzzy correspondence matrix. In essence, it provides the tool to distinguish between Möbius transformations which are close to isometries and those which are general conformal mappings.

At this stage of the Möbius Voting algorithm, we are given two flattened point samples Σ_1 and Σ_2, aligned in the canonical domain by the Möbius transformations m_1, m_2 defined by a triplet of correspondences, and a candidate correspondence $c : \Sigma_1 \to \Sigma_2$ (defined by mutually closest points). The correspondence c can be seen as (possibly local) deformation of one surface onto the other, and our goal is to measure the deviation of this deformation from identity (as we factored out isometry, if it existed, by moving to the canonical domain). In this Section we will define the intrinsic deformation error formula $\mathbf{E}(c)$ used to score the correspondence c.

In the continuous setting, given a conformal flattening $\Phi_1 : \mathcal{M}_1 \to \hat{\mathbb{C}}$ of a smooth surface $\mathcal{M}_1$, scaling is introduced and is coded in the so-called *conformal factors*. These are scaling of the area element induced by the conformal flattening Φ_1. We denote the conformal factors by a non-negative function κ_1. Similarly, denote by κ_2 the conformal factors induced by the flattening Φ_2 of surface $\mathcal{M}_2$. Let us also refer to κ_1 and κ_2 as *densities*. κ_1 and κ_2 in addition to the unifomizations Φ_1 and Φ_2 encapsulate the entire intrinsic geometry.

There are several ways to define an intrinsic distortion measure between the surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$ based on the conformal factors. For example, Jin et al.[2004] and Zeng et al.[2008] define the distortion as the L_2 difference of the conformal factors. However, in our case the distance measure should *not* be a "regular" L_p norm. The reason is that measuring the difference between densities in L_p norm does
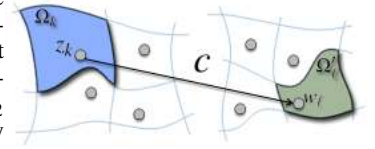
not consider spatial-neighboring relations between points, instead it only compares the density point-wise [Rubner et al. 2000].

Instead, we suggest computing a "transportation-type" cost function. Let us first explain this suggested deformation error in the continuous setting and afterwards give the discretization we actually use. Assume we are given two surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$, their conformal flattening maps Φ_1, Φ_2 and their corresponding conformal factors $\kappa_1(z), \kappa_2(z), z \in \hat{\mathbb{C}}$. We want to score a potential correspondence (deformation) $\tilde{c} : \hat{\mathbb{C}} \to \hat{\mathbb{C}}$ which takes density κ_1 onto density κ_2 by asking how much work does this correspondence put in order to move density κ_1 to density κ_2. That is $\tilde{c}_*(\kappa_1) = \kappa_2$, where $\tilde{c}_*$ is the push forward of density which means that for every subset $D \subset \hat{\mathbb{C}}$ there exists $\int_{\tilde{c}^{-1}(D)} \kappa_1(z)dxdy = \int_D \kappa_2(w)dxdy$, where $dxdy$ is the standard area element in the plane. Then the transportation effort of $\tilde{c}$ in taking κ_1 to κ_2 is defined by

$$\mathbf{E}(\tilde{c}) = \int_{\hat{\mathbb{C}}} dist\,(z, \tilde{c}(z))\,\kappa_1(z)dxdy, \tag{5}$$

where *dist* is a distance measure between two points in the complex plane. The actual choice of the distance measure $dist(\cdot, \cdot)$ deserves an explanation. There are two simple choices for a distance measures: the standard Euclidean distance in $\mathbb{C}$: $dist(z, \tilde{c}(z)) = |z - \tilde{c}(z)|$ and the spherical distance (measuring distance on the sphere): $dist(z, \tilde{c}(z)) = \frac{|z - \tilde{c}(z)|}{|1 + \bar{z}\tilde{c}(z)|}$. However, none of these is a natural distance measure on the flattened space. Nevertheless, since after factoring out the nearly isometric Möbius transformations we expect small distances we believe these choices to be practically equivalent to any other choice. We have experimented with both and since they did not seem to give different results we eventually used the slightly simpler Euclidean distance.

Let us now discretize (5) for our needs. As shown in the image on the right, we define the transportation cost of a candidate discrete correspondence set $c : \Sigma_1 \to \Sigma_2$ (created by taking mutually closest points after Möbius normalization) by summing the cost of each pair $(z_k, c(z_k) = w_\ell)$. The cost of a pair (z_k, w_ℓ) is defined as the *work* needed to take z_k to w_ℓ which is the distance between the points $dist(z_k, w_\ell)$ times the "mass" of the point moved.

In our case, every point z_k represents a cell Ω_k in a partition of the surface ($\mathcal{M}_1 = \cup_k \Omega_k$). Since $area(\Omega_k) = \int_{\Phi_1(\Omega_k)} \kappa_1 dxdy$, the total work of a candidate discrete correspondence c would be, following a piecewise constant discretization of the continuous energy (5): $\int_{\hat{\mathbb{C}}} dist(z, \tilde{c}(z))\kappa_1(z)dxdy =$

$$\sum_k dist(z_k, c(z_k)) \int_{\Phi_1(\Omega_k)} \kappa_1(z)dxdy = \sum_k dist(z_k, c(z_k))area(\Omega_k),$$

However, by construction (Section 4), every cell has equal area (importance), and thus the constant $area(\Omega_k)$ can be dropped to get:

$$\mathbf{E}(c) = \sum_k dist(z_k, c(z_k)) = \sum_k |z_k - c(z_k)|, \tag{6}$$

and the average of this cost, that is $\mathbf{E}(c)$ divided by the number of pairs defined by c is the value of our deformation error.

8 Correspondence matrix processing

The final stage in our algorithm is the processing of the fuzzy correspondence matrix C to produce a discrete set of correspondences (z_j, w_j) with a confidence score $S_j \geq 0$ for each one. Let us normalize C by its maximal value to have all its values in $[0, 1]$. One natural option would be to solve this problem globally, that is solving a linear assignment problem which finds a maximal assignment. However, since it solves the problem globally, it optimize over *all* possible correspondences including bad ones. Since we cannot expect to have *all* (or even most of) the points in Σ_1, Σ_2 to correspond, this method is less reasonable in our context.

Instead, we suggest a greedy algorithm that iteratively looks for the maximal entry greater than zero in the matrix C, say $C_{i,j}$, then adds the correspondence pair (z_i, w_j) to the list of correspondences and gives it the cost $S_j := C_{i,j}$. Then the algorithm zero out the row and column of that element, that is the i-th row and j-th column, and repeats. The algorithm stops when there are no values in C greater than zero. Let us call this algorithm max-row-column algorithm. A user desired value of confidence $S \in [0, 1]$ is then used for taking all correspondences with confidence equal or higher than S. Denote this set by (z_j, w_j), where $j = 1, 2, ..., Q$.

Usually high confidence values indicates high quality correpondences. However, it is possible that there will be good pairs of correpondences with low confidence score. This is mainly due to the fact that the amount of voting each isometric patch receives is directly related to its *area* on the surface mesh. Therefore, it is worthwhile to further search the correspondence list for good pairs with low confidence values. For that end we use the correpondences with high confidence value to further search the list of correspondences outputted from the max-row-column algorithm. Let us assume from this point that the list of correspondences (z_j, w_j) is sorted by the confidence values S_j, such that $S_1 \geq S_2 \geq ...$ Then, the search is done in the following way: for each pair (z_j, w_j) with confidence lower than the desired threshold, we represent z_j and w_j as feature vectors of the normalized geodesic distances to the set of good correspondences. That is, denote by $g(p_j, p_i)$ the geodesic distance on mesh $\mathcal{M}_1$ of the points $p_j, p_i \in \Sigma_1$ (remember that $z_j = \Phi_1(p_j)$). Then we define $a_j = \big(g(p_j, p_1), g(p_j, p_2), ..., g(p_j, p_Q)\big)$, and then set the feature vector of z_j to be $a(z_j) := a_j / median(a_j)$, that is the normalized geodesic distances to the "good set" normalized by the median of this vector. The average is taken so to be robust to surfaces which are not purely isometric. Similarly, we define the feature vector $b(w_j)$ using geodesic distances on $\mathcal{M}_2$. Then we look for closest feature vector $b(w_\ell)$ to the vector $a(z_j)$ using a relative L_2 distance, that is $\|a(z_j) - b(w_\ell)\|^2 = \frac{\|a(z_j) - b(w_\ell)\|_{L_2}^2}{\|a(z_j)\|_{L_2} \|b(w_\ell)\|_{L_2}}$, where $\| \cdot \|_{L_2}$ is the usual L_2 norm. If $g(w_\ell, w_j) < \delta$ where δ is a small parameter (say around 0.05 of the surface diameter) and also in a symmetric manner also $g(z_k, z_j) < \delta$ where z_k is closest in feature space to w_j then we add the pair (z_j, w_j) to our list of final correspondences.

9 Results

In this section, we present results of experiments with the algorithms described in the preceding sections. Our goal is to investigate the range of inputs for which they are able to find good point correspondences, and thus our discussion starts with easier cases and ends with harder ones. All experiments were run on a 2.2GHz Opteron 275 processor.

Non-Rigid World Benchmark: For our first experiment, we tested our methods on a set of 3D models that are part of a Non-Rigid World Benchmark soon to be distributed by Bronstein, Bron-
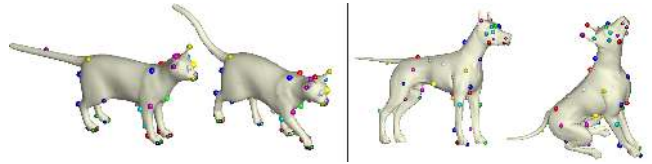

Figure 7: Two examples from the Non-Rigid World Benchmark by Bronstein, Bronstein, & Kimmel [2008].

stein, & Kimmel [2008]. This data set contains several examples where meshes with the same topology are available for the same object in multiple poses (Figure 7), making it possible to predict point correspondences with an automatic algorithm and then measure how close they are to "ground truth". Figure 8 shows our results for processing one cat vs. all other cats, one dog vs. all other dogs, and one wolf vs. all other wolves. Roughly speaking, if we consider less than 3% error acceptable for an application (e.g., morphing), then all first 45 good correspondences are found in all but two examples. If higher error is acceptable (e.g., for similar shape matching), then perhaps 50 or more correspondences are within tolerance. Note that all the correspondences shown in Figure 8,9 are direct output of the voting algorithm *without* the further correspondence matrix processing described in the last paragraph of Section 8 (except where indicated explicitly in Figure 9).

SCAPE Data Set: In a second experiment, we tested our algorithm on the SCAPE data set [Anguelov et al. 2005], which contains meshes of a single person in different poses reconstructed from range images. This data set is slightly harder than the previous one because the meshes are reconstructed from scans, and thus different instances do not have exactly the same local shape features. However, the reconstructions are derived from a template, and thus provide consistent topology, making quantitative comparison of predicted correspondences to ground truth possible. Figure 9 shows the results of finding correspondences between one person (shown in the top left) and all the others. In this case, all of the top 15 correspondences have less than 3% error in all but five of the examples, and many have more than 30 such correspondences. One failure case is shown in the bottom-right of the figure (its row is marked by a orange square in the colored error matrix) – in this case, the front of one person was mapped onto the back of the other.

SHREC 2007 Watertight Benchmark: The third experiment tests our algorithms on meshes of different objects within the same semantic class. For this test, we used the SHREC 2007 Watertight Benchmark Data Set [Giorgi et al. 2007], which contains watertight meshes for 20 different objects with 20 different object classes. Figures 13 and 11 show representative results for the "Human" and "Four-Legged" classes. For these cases, ground truth correspondences are not available, and thus we show only visualizations of correspondences predicted with the highest confidence. For example, looking at the bulldog↔cow in the top-left of Figure 11, we see that correspondences are found at the tips of the noses, throughout the torso, on the "knees" of the legs, etc. Similarly, looking at Figure 13, we see that many good correspondences are found between people with very different body shapes and in different poses. In particular, note that good correspondences are found even in the cases where an input mesh has genus three (i.e., the hands and the legs are joined in the sitting woman fifth from the left in the top row). There are some poor correspondences (e.g., on the legs of the cow↔horse), but they are usually the ones with least confidence. There are also some extreme failure cases (e.g., cow↔giraffe). However, overall, the results are quite positive – similarities in the the global intrinsic structures of the surfaces guide the algorithm to
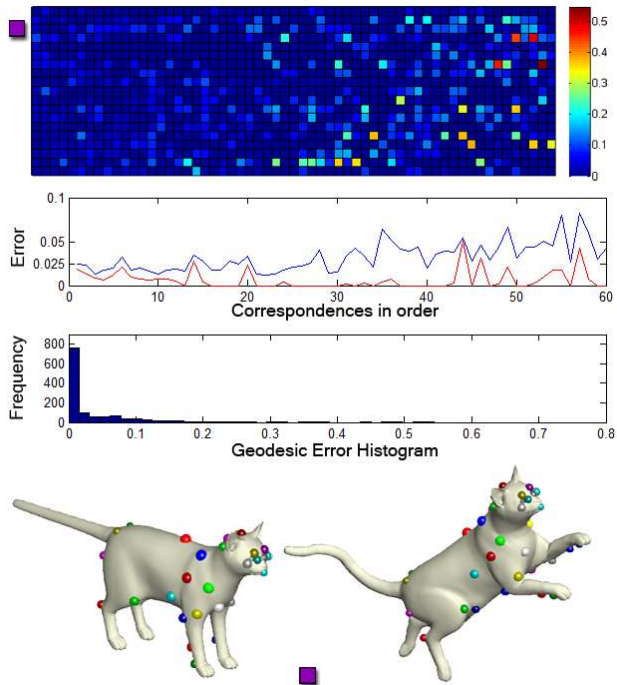
Figure 8: Results of finding corresponding points in the cats, dogs and wolves meshes of the Non-Rigid World Benchmark [Bronstein et al. 2007]. The images on the bottom of the figure show an example result of our algorithm for the cat in two different poses. In the top of the figure there is a colored error matrix, where each row represents a pair of meshes (the example pair of cats appear in the row indicated by a purple square), and each column represents a point correspondence, sorted left-to-right by predicted confidence. The color of each matrix element indicates error from ground truth, reported as geodesic distance between predicted and actual correspondences normalized by the geodesic diameter of the mesh. A plot of this error (mean in blue curve and median in red curve) for the most confident 60 correspondences and a histogram of its distribution are shown at the bottom of the figure.

establish good correspondences between meshes with significantly different global and local shapes. This result is consistent across several other sets of models. For example, Figure 10.

Different Object Types: The next case shows results for the particularly hard case of matching a man to a gorilla, motivated by the example in Figure 12.2 of [Bronstein et al. 2008] (Figure 12). Even though the two surfaces are very far from isometric, and no landmarks are used to guide the search for correspondences, the algorithm is able to match many semantic features correctly (e.g., nose to nose, ears to ears, fingers to fingers, etc.).

Partial Matching Example: In another experiment, we demonstrate that the algorithm is able to establish correspondences between surfaces that are only partially and approximately isometric. Figure 12 shows images of correspondences found for horse↔centaur. This result is possible with direct application of our algorithm because we accumulate votes for correspondences rather than for aligning transformations – the piecewise nature of the correspondence matrix is discovered automatically during the voting stage, without a separate step for transformation clustering.
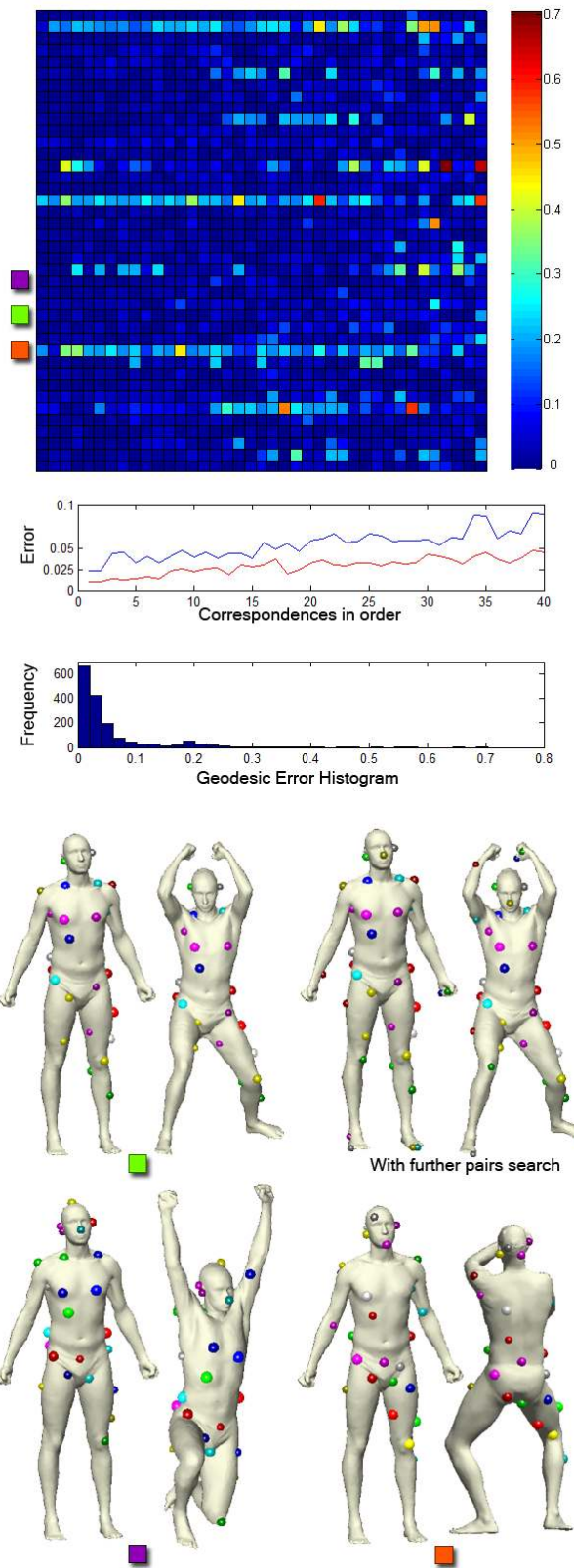
Figure 9: Results of finding corresponding points between one mesh (shown in the top-left) and all others in the SCAPE database. The layout and color-coding of this figure is similar to Figure 8 – please refer to its caption for details.
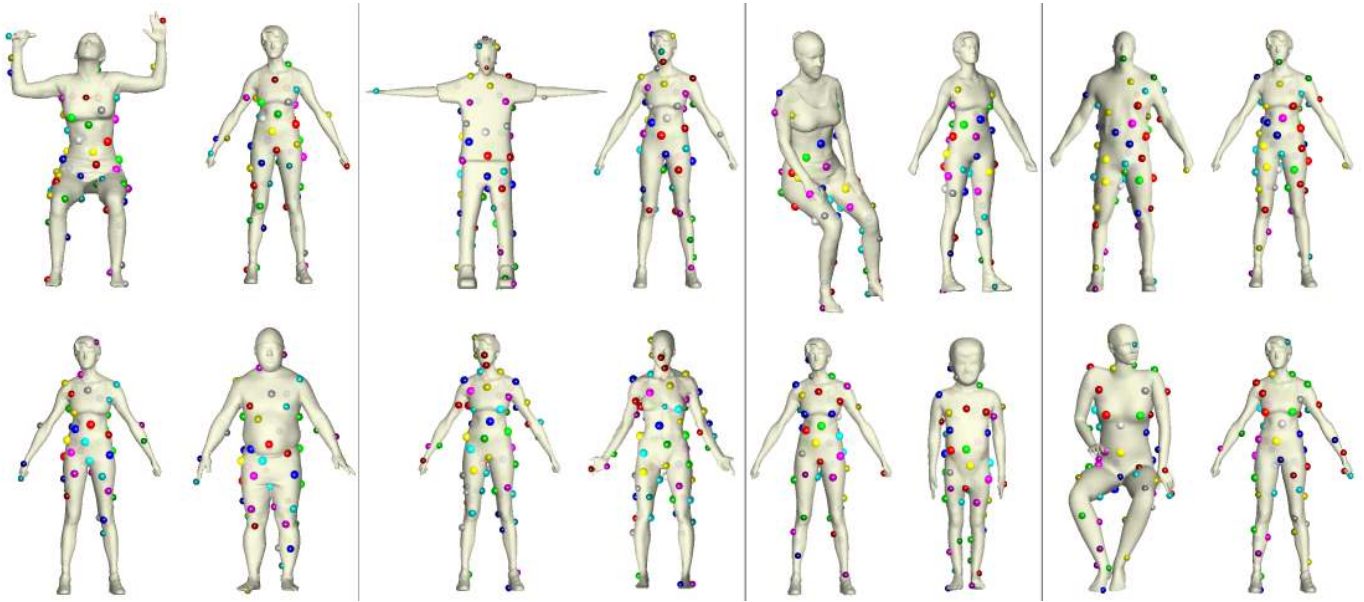
Figure 13: Correspondences found on human models from the SHREC 2007 Watertight Benchmark.
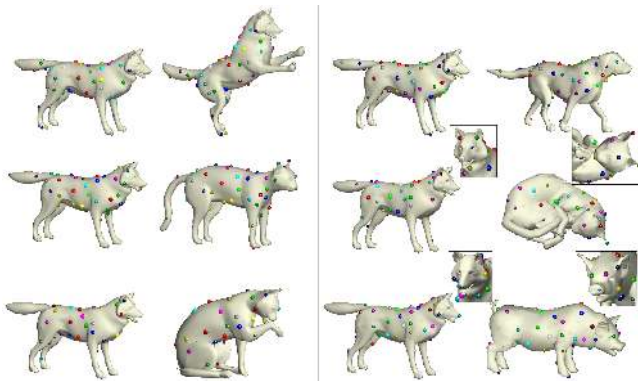

Figure 10: Example of comparing a wolf to itself and to other four legged animal models in different poses.
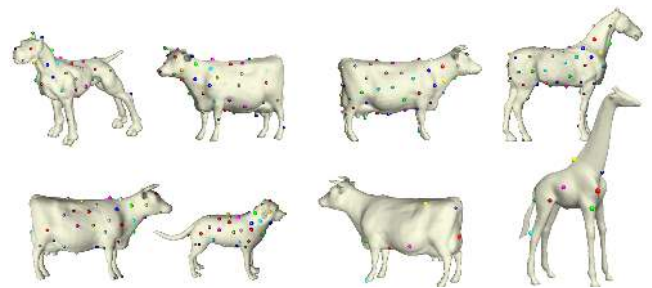

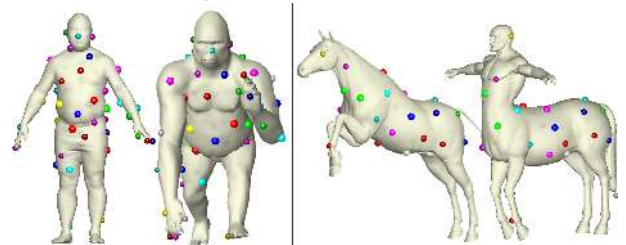Figure 11: Correspondences found on four legged animals from the SHREC 2007 Watertight Benchmark.

Figure 12: Correspondences found automatically between meshes of a gorilla and a man (left), and meshes of a centaur and a horse (right).

Comparison to Previous Work: In a final experiment, we compare our method to the related work [Zhang et al. 2008] (Figure 14). In this case, we find sixty "good"correspondences, while Zhang et al. find five and mismatch the arms. The reasons are mainly that their algorithm searches for correspondences only amongst a sparse set of points, due to the complexity of combinatorial search, while we consider a larger set of points and have a more flexible deformation error.

Computational Complexity and Running Time: Our system has two stages, mid-edge uniformization and Möbius Voting. The first stage is relatively simple – it is equivalent to solving a sparse linear (Laplacian) system. During this stage, we currently calculate the cutting face via an average geodesic distance calculation at each vertex, which is relatively slow ($O(n^2 \log(n))$ for n vertices). However, this is an implementation detail, and we believe that much faster methods can be chosen for this relatively simple step, which has little or no affect on the quality of the results.

The main complexity of our algorithm is in the Möbius Voting stage. It has expected computational complexity of $O(N^4 \log(N))$, where N is the number of points in each point set. For each

of $O(N^3)$ random triplets, computing the Möbius transformations takes constant time, and computing the deformation error takes $O(N \log(N))$ (it depends on finding $O(N)$ closest points, each of which can be found in $O(\log(N))$ with a kd-tree or other spatial indexing structure). Note that this running time is polynomial in the size of the point set, N, and it is independent of the number of vertices in the mesh.

Of course, in practice, the running time depends linearly on the number of iterations. In order to acquire strong evidence for "correct" correspondences, we generally aim to cast ~ 10 votes for each "correct" triplet of correspondences, which means that the to-
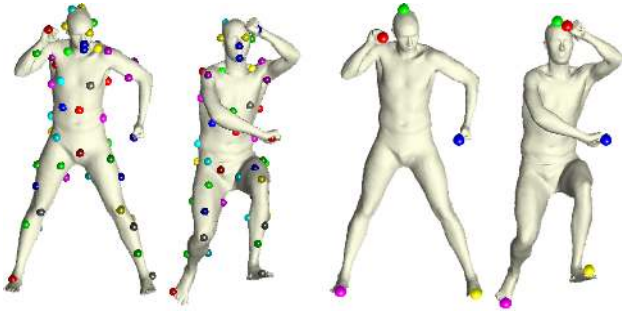
Figure 14: Comparison with the method of [Zhang et al. 2008]. On the left: our Möbius voting result where 61 correspondences found. On the right: Zhang et al. method is applied to five points. Note that sparse sampling is not always enough to find the correct matches.

tal number of iterations should be approximately $\sim 10 \times N^3$ (one out of every N^3 triplets of correspondences is "correct", on average). In all the examples in the paper, we took $80 - 120$ sampled points (N) and randomized between $5M$ and $20M$ triplets. The empirical running times are 70 seconds per $1M$ votes for $N = 80$, and 160 seconds per $1M$ votes for $N = 120$.

10 Discussion

The main contribution of this paper is the idea of leveraging the low-dimensionality of the Möbius group in a voting algorithm that finds correspondences automatically. Our results suggest that it is able to find dozens of good correspondences not only between meshes that are nearly isometric (the same object in different poses), but also between different objects (dog↔wolf), partial matches (horse↔centaur), and even between meshes that are not locally similar at all (man↔gorilla).

This paper represents just a first step, and there are many limitations that suggest directions for future work.

First, the main drawbacks of our current implementation are: 1) the use of a linear flattening technique, which, although fast and simple to compute, suffers from bad artifacts when the meshes in consideration have bad triangles; and 2) the use of a average geodesic distance calculation to find a cutting face for mid-edge uniformization, which although robust, is far too slow for this simple task. We believe that the quality and/or timing of our results could improve with selections of different algorithms for these components.

Second, the algorithm is designed only for topological spheres with holes. Interestingly, good correspondences can be found even when one or both meshes have handles (such an example appeared in the SHREC watertight set). However, this is not guaranteed. Extending the theory behind the method to higher genus objects is a topic for future work.

Third, our algorithm tends to find more correspondences in larger regions of the surfaces (e.g., on the body) – this is because the total magnitude of votes for a correspondence is related to the size of the corresponding surface patch within which it resides and our selection of a setting for the parameter K, the number of mutually closest correspondences needed to log a vote (usually 40% in our examples). To overcome this problem, we believe it would be possible to implement an adaptive, hierarchical algorithm, where spatial relationships to high confidence votes found early in the process are used to corroborate votes for lower confidence ones, and/or where the process proceeds coarse-to-fine by recursively breaking the problem into smaller and smaller surface patches.

Finally, there are a number of applications that might benefit from the correspondences generated by the algorithm. While some of the correspondences are not perfect (average errors are shown in Figures 8 and 9), they are probably good enough to be used directly for shape matching, morphing, symmetry analysis, statistical analysis, and so on. Studying how the needs of these applications interact with the capabilities of automatic correspondence finding and outlier rejection is an interesting topic for further work.

Proof of the mid-edge boundaries property. Let us prove Theorem 5.1. We make use of notations introduced in Section 5. Given a discrete harmonic piecewise linear and continuous function on the mesh $u = \sum_i u_i \phi_i(\cdot)$, for each vertex $v_i \in V$ excluding those of the face we cut out there exists

$$\int_{D_i} \langle \nabla \phi_i, \nabla u \rangle dvol = 0, \qquad (7)$$

where D_i is the 1-ring of vertex i. This can be seen by differentiating the discrete Dirichlet integral $E_D(u) = \sum_{f \in F} \int_f |\nabla u|^2 dvol$ with respect to u_j.

Next, consider a boundary vertex v_j of the mesh $\mathcal{M}$. Denote by $\mathbf{v}_r, \mathbf{v}_s$ the mid-edge vertices on the two boundary edges touching vertex v_j. We will show that $u^*(\mathbf{v}_r) = u^*(\mathbf{v}_s)$ and this will imply the theorem since u^* is defined as the Y-coordinate embedding of the mid-edge vertices (see Figure 4). From the integrability property of the conjugate gradient field $\nabla u^* = J \nabla u$ (see [Polthier 2005], Proposition 35), it is known that $u^*(\mathbf{v}_r) - u^*(\mathbf{v}_s) =$

$$\int_\gamma du^* = - \sum_{f \subset D_j} \langle \nabla |_f u, \nabla |_f \phi_j \rangle area(f) = - \int_{D_j} \langle \nabla u, \nabla \phi_j \rangle dvol,$$
$$(8)$$

where γ is the piecewise linear path starting at $\mathbf{v}_r$ and passing through the mid-edge vertices of the 1-ring neighborhood of v_j ending at $\mathbf{v}_s$, and $f \subset D_j$ means traversing all the faces touching vertex v_j. Now from Eq. (7) it is clear that the last term on (8) equals zero, completing the proof.

11 Acknowledgements

References

ALLEN, B., CURLESS, B., AND POPOVIC, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM Press, New York, NY, USA, 587–594.

ANGUELOV, D., SRINIVASAN, P., PANG, H., KOLLER, D., THRUN, S., AND DAVIS, J. 2004. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*, vol. 17, 33–40.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. *ACM Trans. on Graphics 24*, 3, 408–416.

AUDETTE, M. A., FERRIE, F. P., AND PETERS, T. M. 2000. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis 4*, 3, 201–217.

BALLARD, D. H. 1981. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition 13*, 2, 111–122.

BERG, A. C., BERG, T. L., AND MALIK, J. 2005. Shape matching and object recognition using low distortion correspondence. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2, 239–256.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1, 213–230.

BRONSTEIN, A., BRONSTEIN, M., AND KIMMEL, R. 2006. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Science*, 1168–1172.

BRONSTEIN, A. M., BRONSTEIN, M. M., , AND KIMMEL, R. 2007. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Trans. Visualization and Computer Graphics 13*, 5, 902–913.

BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2008. *Numerical Geometry of Non-Rigid Shapes*. Springer.

BROWN, B. J., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-d scans. *ACM Trans. Graph. 26*, 3, 21.

CHUI, H., AND RANGARAJAN, A. 2003. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst. 89*, 2-3, 114–141.

DESBRUN, M., MEYER, M., SCHRDER, P., AND BARR, A. H. 2002. Discrete differential-geometry operators for triangulated 2-manifolds. Springer-Verlag, 35–57.

ECKSTEIN, I., PONS, J., TONG, Y., KUO, C., AND DESBRUN, M. 2007. Generalized Surface Flows for Mesh Processing. In *Symposium on Geometry Processing*, 183 – 192.

ELAD, A., AND KIMMEL, R. 2003. On bending invariant signatures for surfaces. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 10, 1285–1295.

ELDAR, Y., LINDENBAUM, M., PORAT, M., AND ZEEVI, Y., 1997. The farthest point strategy for progressive image sampling.

FUNKHOUSER, T., AND SHILANE, P. 2006. Partial matching of 3d shapes with priority-driven search. In *Symposium on Geometry Processing*.

GELFAND, N., MITRA, N. J., GUIBAS, L. J., AND POTTMANN, H. 2005. Robust global registration. In *Symposium on Geometry Processing*.

GIORGI, D., BIASOTTI, S., AND PARABOSCHI, L., 2007. SHREC:SHape REtrieval Contest: Watertight models track, http://watertight.ge.imati.cnr.it/.

GOLD, S., RANGARAJAN, A., PING LU, C., AND MJOLSNESS, E. 1998. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition 31*, 957–964.

HORMANN, K., LVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice. In *ACM SIGGRAPH Course Notes*.

HUANG, Q., ADAMS, B., WICKE, M., , AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. In *Proc. of Eurographics Symposium on Geometry Processing 2008 (SGP)*, 1149–1458.

JAIN, V., ZHANG, H., AND VAN KAICK, O. 2007. Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling 13*, 1, 101–124.

JIN, M., WANG, Y., YAU, S.-T., AND GU, X. 2004. Optimal global conformal surface parameterization. In *VIS '04: Proceedings of the conference on Visualization '04*, IEEE Computer Society, Washington, DC, USA, 267–274.

JOHNSON, A., AND HEBERT, M. 1999. Using spin-images for efficient multiple model recognition in cluttered 3-D scenes. *IEEE PAMI 21*, 5, 433–449.

KRAEVOY, V., AND SHEFFER, A. 2006. Mean-value geometry encoding. *International Journal of Shape Modeling 12*, 1.

LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. In *Symposium on Geometry Processing*.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph. 24*, 3, 479–487.

LOWE, D. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2, 91–110.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2007. Symmetrization. In *ACM Transactions on Graphics (SIGGRAPH)*, vol. 26.

OVSJANIKOV, M., SUN, J., AND GUIBAS, L. 2008. Global intrinsic symmetries of shapes. In *Eurographics Symposium on Geometry Processing*, vol. 27.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. 2005. Example-based 3D scan completion. In *Symposium on Geometry Processing*, 23–32.

PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 15–36.

POLTHIER, K. 2005. Computational aspects of discrete minimal surfaces. *Global Theory of Minimal Surfaces, Proc. of the Clay Mathematics Institute 2001 Summer School, David Hoffman (Ed.), CMI/AMS*.

PRAUN, E., AND HOPPE, H. 2003. Spherical parametrization and remeshing. *ACM Trans. Graph. 22*, 3, 340–349.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision 40*, 2, 99–121.

RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*, 145–152.

SHEFFER, A., GOTSMAN, C., AND DYN, N. 2004. Robust spherical parameterization of triangular meshes. *Computing 72*, 1-2, 185–193.

SPRINGER, G. 1981. *Introduction to Riemann Surfaces*. AMS Chelsea Publishing.

SUMNER, R., AND POPOVIC, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. on Graphics (SIGGRAPH) 23*, 3, 399–405.

WANG, S., JIN, M., AND GU, X. D. 2007. Conformal geometry and its applications on 3d shape matching, recognition, and stitching. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 7, 1209–1220. Member-Yang Wang and Member-Dimitris Samaras.

ZENG, W., ZENG, Y., WANG, Y., YIN, X., GU, X., AND SAMARAS, D. 2008. 3d non-rigid surface matching and registration based on holomorphic differentials. *The 10th European Conference on Computer Vision (ECCV)*.

ZHANG, H., SHEFFER, A., COHEN-OR, D., ZHOU, Q., VAN KAICK, O., AND TAGLIASACCHI, A. 2008. Deformation-driven shape correspondence. *Comput. Graph. Forum 27*, 5, 1431–1439.